Optical Engineering

SPIEDigitalLibrary.org/oe

Adaptive search range adjustment scheme for fast motion estimation in AVC/H.264

Sunyoung Lee
Kiho Choi
Euee S. Jang

Optical Engineering 50(6), 067402 (June 2011)

Adaptive search range adjustment scheme for fast motion estimation in AVC/H.264

Sunyoung Lee
Kiho Choi
Euee S. Jang
Hanyang University
Digital Media Laboratory
Haengdang-dong 17
Seongdong-gu
Seoul 133-791, Republic of Korea
E-mail: esjang@hanyang.ac.kr

Abstract. AVC/H.264 supports the use of multiple reference frames (e.g., 5 frames in AVC/H.264) for motion estimation (ME), which demands a huge computational complexity in ME. We propose an adaptive search range adjustment scheme to reduce the computational complexity of ME by reducing the search range of each reference frame—from the (*t*-1)'th frame to the (*t*-5)'th frame—for each macroblock. Based on the statistical analysis that the 16×16 mode type is dominantly selected rather than the other block partition mode types, the proposed method reduces the search range of the remaining ME process in the given reference frame according to the motion vector (MV) position of the 16×16 block ME. In the case of the (*t*-1)'th frame, the MV position of the 8×8 block ME—in addition to that of 16×16 block ME—is also used for the search range reduction to sub-block partition mode types of the 8×8 block. The experimental results show that the proposed method reduces about 50% and 65% of the total encoding time over CIF/SIF and full HD test sequences, respectively, without any noticeable visual degradation, compared to the full search method of the AVC/H.264 encoder. [DOI: 10.1117/1.3589292]

Subject terms: search range; motion estimation; video coding; AVC/H.264.

Paper 100733RR received Sep. 8, 2010; revised manuscript received Apr. 5, 2011; accepted for publication Apr. 13, 2011; published online Jun. 6, 2011.

1 Introduction

The latest video coding standard, MPEG-4 AVC/H.264, adopts variable block size and multiple reference frames for motion estimation (ME) to find the best matched block(s) in a given search range, at the cost of high computational complexity.[1] In order to reduce the computational complexity of ME, many have tried to reduce the search range using spatial and/or temporal motion vector (MV) information. In Ref. 2, the size and direction of a MV predictor are used to adjust the search range. An adaptive search range algorithm based on MV difference is proposed in Ref. 3. In Ref. 4, the initial search zones are determined by block type and reference frame number and then the search range is set on the basis of spatial MVs and MVs of four 8×8 blocks. In Ref. 5, two prediction methods (i.e., MEDIAN and UP_LAYER) to calculate MV predictors are used to adjust the search range for integer ME. Another approach to reduce the search range is introduced by exploiting the sum of absolute difference (SAD) value in Ref. 6. Lu et al.[7] propose a method to predict the search range using the image size, the block mode types, and the quantization parameters (Qps). Many tried to design the search range reduction scheme with variable block sizes in AVC/H.264. However, there has been very little effort in designing an efficient search range scheme for multiple reference frames.

In this paper, we propose an approach to adaptively adjust the search range for ME over multiple reference frames in AVC/H.264. It is generally true that the shorter the temporal distance between the reference and current frames, the stronger the correlation between the frames as shown in Table 1. In other words, the previous [or (*t*-1)'th] reference frame to the current (or *t*'th) frame is selected more frequently than the other reference frames. Although Table 1 is an example based on Foreman sequence, the tendency of the selection holds in most cases. From this observation, the proposed method applies two levels of the search range adjustment: the one for the (*t*-1)'th frame and the other for the (*t*-2)'th through the (*t*-5)'th frames.

For the (*t*-2)'th through the (*t*-5)'th reference frames, the search range adjustment is based on the MV position of the 16×16 block motion estimation (16×16ME) of the given reference frame. As shown in Table 2, the MV position of 16×16ME is dominantly selected rather than those of the sub-partition blocks (i.e., 16×8, 8×16, and P8×8) in the (*t*-2)'th through the (*t*-5)'th frames because a few of the ME results in sub-partition blocks are better than that of 16×16ME in a rate-distortion (RD) sense.

In the case of the (*t*-1)'th reference frame, the proposed method additionally uses the MV position of the 8×8 block motion estimation (8×8ME) to adjust the search range of the sub-partition blocks (i.e., 8×4, 4×8, and 4×4), along with that of 16×16ME for the sub-partition blocks (i.e., 16×8 and 8×16). It is due to the fact that the MV position of 8×8ME is better suited than that of 16×16ME for the sub-partition blocks (i.e., 8×4, 4×8, and 4×4). The two-level approach in the search range adjustment is necessary because the P8×8 block types are selected more in the (*t*-1)'th reference frame than in the (*t*-2)'th through the (*t*-5)'th frames for the sequences with high motion activity or sequences that are encoded in low quantization parameter value as shown in Table 3. Through the experimental results, we show that the proposed method outperforms the full search range (FSR) method of JM 11.0 in computation time at the similar RD cost.

0091-3286/2011/$25.00 © 2011 SPIE

Table 1 The selection ratio of five reference frames in the Foreman sequence.

Qp	Reference frame number				
	(t-1)'th	(t-2)'th	(t-3)'th	(t-4)'th	(t-5)'th
22	64.96	13.77	10.51	5.44	5.32
28	75.43	10.37	7.83	3.21	3.15
34	88.12	5.64	3.81	1.12	1.31
Average	**76.17**	**9.93**	**7.38**	**3.26**	**3.26**

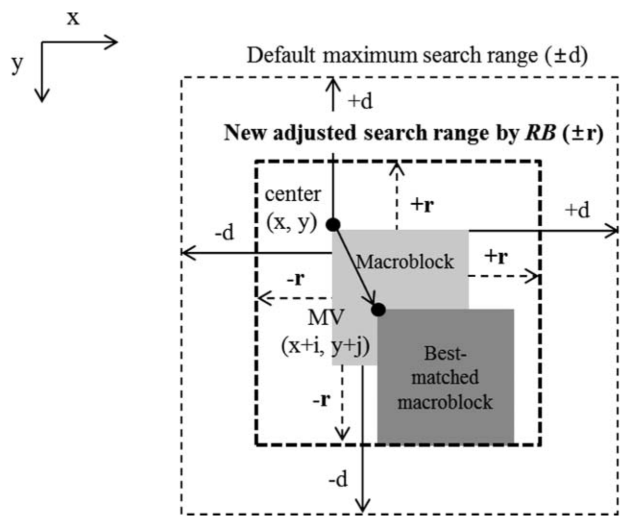

Fig. 1 Search range adjustment using the range boundary.

2 Proposed Method

In order to find the best matched blocks during ME, a rate-distortion function is usually used in the AVC/H.264 encoding process. More specifically, the MV and the reference frame (REF) for each block partition type (i.e., 16×16 to 4×4) are determined in the AVC/H.264 JM encoder through the following evaluation:

$$J_{\text{motion}} = \text{SAD} + \lambda_{\text{motion}} \times R(\text{MV}, \text{REF}) \tag{1}$$

where SAD is the sum of the absolute difference between source block and reconstructed block with MV and REF, λ_{motion} the Lagrangian multiplier, and R(MV, REF) the bit-rates for coding MV and REF. Let the RD cost of 16×16ME be

$$J_{16\times 16\text{ME}} = \text{SAD}_{16\times 16} + \lambda \times R_{16\times 16}(\text{MV}_{16\times 16}, \text{REF}_{16\times 16}). \tag{2}$$

For a 16×16 macro block (MB), there is one 16×16ME, while there are two 16×8 block motion estimations (16×8MEs) for the upper 16×8 and lower 16×8 blocks. Therefore, the RD cost of 16×8 partition type for an MB during ME can be rewritten as follows:

$$\begin{aligned} J_{16\times 8} &= J^{\text{Up}}_{16\times 8\text{ME}} + J^{\text{Low}}_{16\times 8\text{ME}} \\ J^{\text{Up}}_{16\times 8\text{ME}} &= \text{SAD}^{\text{Up}}_{16\times 8} + \lambda \times R^{\text{Up}}_{16\times 8}\left(\text{MV}^{\text{Up}}_{16\times 8}, \text{REF}^{\text{Up}}_{16\times 8}\right) \\ J^{\text{Low}}_{16\times 8\text{ME}} &= \text{SAD}^{\text{Low}}_{16\times 8} + \lambda \times R^{\text{Low}}_{16\times 8}\left(\text{MV}^{\text{Low}}_{16\times 8}, \text{REF}^{\text{Low}}_{16\times 8}\right). \end{aligned} \tag{3}$$

Assuming that $\text{SAD}_{16\times 16}$ for 16×16ME is close to the sum of $\text{SAD}^{\text{Up}}_{16\times 8}$ and $\text{SAD}^{\text{Low}}_{16\times 8}$ for 16×8MEs, R(MV, REF) becomes a dominant factor in the RD function. If the 16×8 partition type is selected instead of 16×16, the bit-rates for describing 16×8 should be lower than that of 16×16. In other words, the sizes of two MVs in 16×8 should be smaller than that of MV in 16×16 case in the same reference frame. This analogy can be extended to other sub-partition blocks. Therefore, the MV position of 16×16ME can be used as a threshold value for the search range adjustment of the sub-partition blocks (i.e., 16×8 to 4×4) in the given reference frame. Likewise, the MV position of 8×8ME can also be used as a threshold value to adjust the search range of the sub-partition blocks (i.e., 8×4 to 4×4).

The threshold value is used as the range boundary (RB) to define a new search range that can be set between zero and default maximum search range (i.e., $0 \leq \text{r} \leq \text{d}$) as shown in Fig. 1. In our experiment, we chose the minimum RB value to be one instead of zero. In order to keep the computational complexity low, the RB value should be kept low. On the other hand, a low RB value will decrease the compression efficiency. Therefore, the efficient selection of the RB value is critical in determining the encoder performance in both compression efficiency and computational complexity. In this paper, we determine the RB value with the following equation:

$$\begin{aligned} (\text{MV}_x, \text{MV}_y) &= \arg \min_{(\text{MV}_x, \text{MV}_y)} J_{\text{motion}}(\text{MV}_x, \text{MV}_y) \\ \text{RB} &= \max(|\text{MV}_x|, |\text{MV}_y|). \end{aligned} \tag{4}$$

From Eq. (4), the best MV position that minimizes the RD cost of the given ME block type within the default

Table 2 The selected ME block type distribution in the Foreman sequence [for the (t-2)'th through the (t-5)'th reference frames].

Qp	ME block type			
	16×16	16×8	8×16	P8×8
22	50.86	15.31	16.83	17.00
28	63.54	12.92	14.99	8.55
34	76.07	9.88	10.83	3.22
Average	**63.49**	**12.71**	**14.22**	**9.59**

Table 3 The selected ME block type distribution in the Foreman sequence [for the (t-1)'th reference frame].

Qp	ME block type				
	SKIP	16×16	16×8	8×16	P8×8
22	20.44	26.60	12.37	13.40	27.19
28	41.46	26.54	9.96	10.69	11.36
34	60.58	23.72	6.12	6.13	3.44
Average	**40.83**	**25.62**	**9.48**	**10.07**	**14.00**

Table 4 The selected ME block type for determining the RB value.

Reference frame number	Block partition type						
				P8×8			
	16×16	16×8	8×16	8×8	8×4	4×8	4×4
(t-2)'th to (t-5)'th	16×16ME with $J_{16\times16ME}$			16×16ME with $J_{16\times16ME}$			
(t-1)'th	16×16ME with $J_{16\times16ME}$			8×8ME with $J_{8\times8ME}$			

maximum search range should be determined first. In this paper, we use the same RD cost function (i.e., J_{motion}) for ME in AVC/H.264 for convenience. The RB value can be derived from the best MV position where we set the RB value to be the absolute maximum value between MV_x and MV_y. Once the RB value is determined, the remaining ME process is conducted using the new search range defined by the RB value. Consequently, the search range adjustment is conducted N times per macroblock if there are N reference frames. Theoretically speaking, the best RB value cannot be determined until the RD costs in all ME block types are computed. We can avoid unnecessary ME computations if it is possible to early determine a good *RB* value.

In the proposed method, we determine the RB value from the best MV position of 16×16ME or 8×8ME. Table 4 shows the selected ME block types for determining the RB value. As the table shows, the MV position of 16×16ME is used to compute the RB value for the corresponding block partition type. In the case of the P8×8 block partition types (e.g., 8×8, 8×4, 4×8, and 4×4) for the (t-1)'th frame, the MV position of 8×8ME is used instead of that of 16×16ME. Through the experiment, we found that 8×8ME is the better choice than 16×16ME to produce a good RB value for the P8×8 block partition types of the (t-1)'th frame. Because of the additional use of 8×8ME, the computational complexity of the proposed method in the (t-1)'th frame is slightly increased to preserve the compression efficiency. We summarize the proposed method in pseudo code as shown in Table 5.

We analyzed the 'in-range' ratio (or hit ratio in Ref. 8) to see if the result of ME is within the new search range defined by the given RB. In Table 6, the in-range ratio is computed based on the selected ME block type distribution (in Table 2) for the (t-2)'th through the (t-5)'th reference frames. In the case of the 16×16 block type, all the MV positions are within the given RB by definition. Overall, the in-range ratio is 97.02%, which means that only about 3% is out of the selected RB value derived from the MV position of 16×16ME. In Table 7, the in-range ratio is computed based on the selected ME block type distribution (in Table 3) for the (t-1)'th reference frame. The overall in-range ratio becomes almost 98%, which means that only about 2% is out of the RB.

3 Experimental Results

In order to evaluate the performance, the proposed method was implemented and tested in the AVC/H.264 JM 11.0 encoder.[9] In this experiment, we employed the full search method for ME. Yet, the proposed method can be implemented in any motion estimation method that allows the search range adjustment (e.g., fast full and hexagon methods in AVC/H.264). Four quantization parameters were used in several CIF/SIF and full HD sequences that are used by ISO/IEC MPEG-H high-efficiency video coding adhoc group.[10] Five reference frames are used for CIF/SIF and four for full HD sequences. The default maximum search range was set to ±16 for CIF/SIF and ±64 for full HD sequences

Table 5 The procedure of the proposed adaptive search range method.

Algorithm 1 adaptive search range method

```
for each reference frame do
  if the reference frame is the (t-1)'th frame
    for each MB do
      if ME is 16×16ME
        maximum_search_range=[(2×SR+1) × (2×SR+1)]
        MV_position=integer ME (maximum_search_range)
        16×16_range_boundary=compute_range boundary
        (MV_position)
      else if ME is 16×8ME or 8×16ME
        maximum_search_range=16×16_range boundary
        MV_position=integer ME (maximum_search_range)
      else if ME is 8×8ME
        maximum_search_range=[(2×SR+1) × (2×SR+1)]
        MV_position=integer ME (maximum_search_range)
        8×8_range boundary=compute_range boundary
        (MV_position)
      else
        maximum_search_range=8×8_range boundary
        MV_position=integer ME (maximum_search_range)
      end if
    end
  else
    for each MB do
     if ME is 16×16ME
        maximum_search_range=[(2×SR+1) × (2×SR+1)]
        MV_position=integer ME (maximum_search_range)
        16×16_range boundary=compute_range boundary
        (MV_position)
      else
        maximum_search_range=16×16_range boundary
        MV_position=integer ME (maximum_search_range)
      end if
    end
  end if
end
```

Table 6 The in-range ratio for each ME block type in the Foreman sequence [for the (t-2)'th to the (t-5)'th reference frames].

	ME block type				
	16×16	16×8	8×16	P8×8	
Qp	IN / OUT	IN / OUT	IN / OUT	IN / OUT	In-range ratio (%)
22	50.86 / 0.00	14.79 / 0.52	15.72 / 1.11	15.58 / 1.42	**96.95**
28	63.54 / 0.00	12.26 / 0.67	13.69 / 1.29	7.73 / 0.82	**97.22**
34	76.07 / 0.00	8.96 / 0.93	9.16 / 1.67	2.69 / 0.52	**96.88**
Average	**63.49** / 0.00	**12.00** / 0.70	**12.86** / 1.36	**8.67** / 0.92	**97.02**

IN: in-range and OUT: out-of-range.

Table 7 The in-range ratio for each ME block type in the Foreman sequence [for the (t-1)'th reference frame].

	ME block type					
	SKIP	16×16	16×8	8×16	P8×8	
Qp	IN / OUT	IN / OUT	IN / OUT	IN / OUT	IN / OUT	In-range ratio (%)
22	20.44 / 0.00	26.60 / 0.00	11.52 / 0.85	12.34 / 1.06	26.27 / 0.92	**97.17**
28	41.46 / 0.00	26.54 / 0.00	9.01 / 0.95	9.57 / 1.11	10.99 / 0.37	**97.57**
34	60.58 / 0.00	23.72 / 0.00	5.46 / 0.66	5.37 / 0.76	3.34 / 0.11	**98.47**
Average	**40.83** / 0.00	**25.62** / 0.00	**8.66** / 0.82	**9.09** / 0.98	**13.53** / 0.46	**97.74**

IN: in-range and OUT: out-of-range.

Table 8 Test environment for the AVC/H.264 JM encoder.

Test sequences	CIF/SIF: Akiyo, Bus, Coastguard, Container, Foreman, Hall, Mobile, Mother-daughter, News, Silent, and Stefan Full HD: BasketballDrive, BQTerrace, Cactus, Parkjoy, ParkScene, PeopleOnStreet, and Traffic
Sequence resolution	CIF (352×288), SIF (352×240), and Full HD (1920×1080)
Total frames to be coded	100
JM version	JM 11.0
Profile	CIF (Main profile with level 3) Full HD (Main profile with level 4)
Quantization parameter	22, 28, 34, and 40
MV search range	CIF/SIF (16×16) and Full HD (64×64)
Reference frames	CIF/SIF (5) and Full HD (4)
GOP structure	IPPP
Entropy coding type	CAVLC
ME scheme	Full search
RDO mode	ON (high complexity)

Table 9 The comparison of average N_{SP} and S_{SR} in the Foreman sequence.

		Proposed method		FSR method		
Qp	Reference frame number	N_{SP} – (A)	S_{SR}	N_{SP} – (B)	S_{SR}	A/B (%)
22	(*t*-1)'th	169.15	6.00	1089	16	15.53
	(*t*-2)'th	172.29	6.06	1089	16	15.82
	(*t*-3)'th	247.74	7.37	1089	16	22.75
	(*t*-4)'th	293.71	8.07	1089	16	26.97
	(*t*-5)'th	331.55	8.60	1089	16	30.45
28	(*t*-1)'th	166.62	5.95	1089	16	15.30
	(*t*-2)'th	175.80	6.13	1089	16	16.14
	(*t*-3)'th	251.69	7.43	1089	16	23.11
	(*t*-4)'th	297.97	8.13	1089	16	27.36
	(*t*-5)'th	337.00	8.68	1089	16	30.95
34	(*t*-1)'th	165.05	5.92	1089	16	15.16
	(*t*-2)'th	181.90	6.24	1089	16	16.70
	(*t*-3)'th	253.69	7.46	1089	16	23.30
	(*t*-4)'th	299.65	8.16	1089	16	27.52
	(*t*-5)'th	338.53	8.70	1089	16	31.09
Average						**22.54**

in horizontal and vertical directions due to full HD features such as its resolution. More details on the encoding environment are described in Table 8. The test was conducted on a PC with 2.4 GHz Intel Quad Core CPU and 3.25 GB RAM running Windows XP.

Table 10 Encoder performance evaluation of the proposed method compared to the FSR method in the AVC/H.264 JM encoder (CIF/SIF).

Sequence	ΔB (%)	ΔPSNR (dB)	ΔET (%)	ΔMET (%)
Akiyo	– 0.05	0.00	– 47.51	– 61.64
Bus	1.00	– 0.02	– 39.35	– 44.30
Coastguard	0.24	– 0.01	– 52.49	– 59.03
Container	0.03	– 0.02	– 55.23	– 66.63
Foreman	0.26	– 0.01	– 47.84	– 56.30
Hall	0.83	– 0.02	– 49.37	– 62.32
Mobile	0.22	– 0.01	– 62.16	– 72.50
Mother & daughter	0.13	– 0.02	– 49.21	– 60.63
News	0.01	– 0.02	– 49.21	– 61.97
Silent	0.40	0.00	– 49.76	– 62.16
Stefan	0.52	– 0.01	– 45.26	– 52.25
Average	**0.33**	**– 0.01**	**– 49.76**	**– 59.98**

For an objective comparison, we counted the number (N_{TSP}) of total searching points for each MB as follows:

$$N_{TSP} = N_{SP} \times N_{BP} \times N_{REF} \tag{5}$$

where N_{SP} indicates the number of searching points in the search range, N_{BP} the number of block partitions, and N_{REF} the number of reference frames. N_{SP} is computed as follows:

$$N_{SP} = [(2 \times S_{SR} + 1) \times (2 \times S_{SR} + 1)] \tag{6}$$

where S_{SR} is the size of the search range. In general, N_{TSP} is 223,245 in the JM encoder when N_{SP} is set to 1,089 (i.e., S_{SR} is ± 16), N_{BP} to 41 (e.g., one 16×16, two 16×8, two 8×16, four 8×8, etc.), and N_{REF} to 5. In the proposed method, N_{SP} is the only variable among the three terms in Eq. (5) where the value of N_{SP} can vary from 9 (i.e., S_{SR} is set to ± 1) to 1,089 (i.e., S_{SR} is set to ± 16) for each reference frame based on the MV position of 16×16ME or 8×8ME.

Table 9 shows the average N_{SP} and S_{SR} of the proposed adaptive search range method and the FSR method in the JM encoder for each reference frame. Compared to the FSR method, the proposed method has reduced the required N_{SP} to about 23% of the FSR method (i.e., 77% reduction). From Table 9, it can be observed that the average N_{SP} and S_{SR} of the proposed adaptive search range method increase when the temporal distance between the current and reference frames increases because of the lower temporal correlation.

Table 11 Encoder performance evaluation of the proposed method compared to the FSR method in the AVC/H.264 JM encoder (full HD).

Sequence	ΔB (%)	ΔPSNR (dB)	ΔET (%)	ΔMET (%)
BasketballDrive	0.31	− 0.03	− 65.21	− 66.50
BQTerrace	− 0.04	− 0.02	− 68.02	− 69.57
Cactus	0.38	0.00	− 66.72	− 68.37
Parkjoy	0.32	0.00	− 53.62	− 69.93
ParkScene	0.17	0.00	− 67.72	− 69.16
PeopleOnStreet	0.14	− 0.02	− 68.64	− 70.32
Traffic	0.01	0.00	− 67.58	− 69.71
Average	**0.18**	**− 0.01**	**− 65.36**	**− 69.08**

For evaluating the overall encoder performance of the proposed method, the following performance criteria such as bit-rate increment (ΔB-%), delta peak signal-to-noise ratio (PSNR) (ΔPSNR-dB), encoding-time reduction (ΔET-%), and ME-time reduction (ΔMET-%) are measured and shown in Tables 10 and 11. For the CIF/SIF sequences in Table 10, the encoder employing the proposed method reduces the total encoding time on average 50% and the ME time on average 60% compared to the JM encoder while it shows a marginal increase in bit-rate (i.e., 0.33%) and a decrease in PSNR (i.e., − 0.01 dB). The proposed method works well to the Mobile sequence with high motion activity. In the case of the full HD sequences, the encoder employing the proposed method saves 65% of the total encoding time with only a 0.18% bit-rate increase and a 0.01 dB loss in PSNR as shown in Table 11. The gain in computation time of the proposed method in the full HD sequences was obtained with the larger search range value (i.e., ± 64) than that of the CIF/SIF sequences (i.e., ± 16) and the less number of reference frames (i.e., 4 instead of 5). This implies that the proposed method is well suited for a large value of search range even when the number of reference frames is small. Consequently, it can be reported that the proposed method is applicable for a wide range of sequences regardless of the motion activity and the video resolution.

4 Conclusion

We proposed an adaptive search range adjustment method as a means to further reduce the computational complexity of ME during the encoding process. The proposed method employed the MV position of the 16×16 block motion estimation as a threshold to adjust the search range of the sub-partitions blocks for each reference frame from the (t-2)'th to the (t-5)'th frame. In addition to the MV position of the 16×16 block, that of the 8×8 block motion estimation was also used to adjust the search range of the sub-partition blocks of the P8×8 block for the (t-1)'th reference frame. The test results demonstrated that the encoder using the proposed method reduces the total encoding time about 50% for the CIF/SIF sequences with a negligible coding bit-rate increase and a marginal video quality degradation compared to the AVC/H.264 JM encoder.

Acknowledgments

This work was supported by the National Research Laboratory, Korea.

References

1. T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003).
2. J. Lee, M. Choi, Y. Cho, J. Kim, and W.-K. Cho, "Fast H.264/AVC motion estimation algorithm using adaptive search range," *Proc. ISIC* **2009**, 336–339 (2009).
3. Z. Chen, Q. Liu, T. Ikenaga, and S. Goto, "A motion vector difference based self-incremental adaptive search range algorithm for variable block size motion estimation," *Proc. IEEE ICIP* **2008**, 1988–1991 (2008).
4. T.-Y. Kuo, J.-J. Shu, and H.-J. Lu, "Dynamic search range control for H.264," *Proc. IEEE ISM* **2007**, 116–121 (2007).
5. Z. Chen, Y. Song, T. Ikenaga, and S. Goto, "A dynamic search range algorithm for variable block size motion estimation in H.264/AVC," *Proc. IEEE ICICS* **2007**, (2007).
6. S. Lee, "Fast motion estimation based on search range adjustment and matching point decimation," *IET Image Processing*, **4**(1), 1–10 (2010).
7. X. Lu and C. Xiao, "A new strategy to predict the search range in H.264/AVC," *Proc. IEEE ICME* **2009**, 21–24 (2009).
8. S.-H. Ri, Y. Vatis, and J. Ostermann, "Fast inter-mode decision in an H.264/AVC encoder using mode and lagrangian cost correlation," *IEEE Trans. Circuits Syst. Video Technol.* **19**(2), 302–306 (2009).
9. AVC/H.264 JM 11.0 reference, at http://iphome.hhi.de/suehring/tml/download/old_jm/.
10. "Resolutions of 93rd Meeting," ISO/IEC JTC1/SC29/WG11 N11356, Geneva, CH, (2010).

Sunyoung Lee received her BS degree in computer science and engineering from Soongsil University, Seoul, Korea, in 1996, an MS degree in information and communications engineering, and a PhD degree in electronics and computer engineering from Hanyang University, Seoul, Korea, in 2005 and 2010, respectively. She has been in a post-doctor course of Brain Korea 21 in Hanyang University since 2010. Her research interests include digital signal procession, image and video compression, and MPEG standardization.

Kiho Choi received his BS degree in information and communication engineering and an MS degree in the Department of Communication and Electronics and Computer Engineering from Hanyang University in Seoul Korea in 2008 and in 2010, respectively. He has been working toward his PhD degree in the Department of Communication and Electronic Engineering at the same university. His research interests include video compression, audio compression, and MPEG standardization.

Euee S. Jang received a BS degree from Jeonbuk National University, Korea, and PhD from SUNY at Buffalo, New York. He is an associate professor at the Department of CSE, Hanyang University, Seoul, Korea. His research interests include image/video coding, reconfigurable video coding, and computer graphics objects. He has authored more than 150 MPEG contribution papers, more than 30 journal or conference papers, 35 pending or patented patents, and 2 book chapters. He has received three ISO/IEC Certificates of Appreciation for the contribution in MPEG-4 development. He received the Presidential Award for the contribution in MPEG standardization from the Korean Government.